Střední průmyslová škola a Vyšší odborná škola, Liberec, příspěvková organizace

2D skákačka – Forbidden gem

Maturitní práce

Autor **Jan Seidel**

Obor **Informační technologie**

Vedoucí práce **Ing. Marek Pospíchal**

Školní rok **2023/2024**

Počet stran **21**

Počet slov **2633**



Anotace

Práce pojednává o kompletním vytvoření 2D skákačky vytvořenou v herním enginu Unity. Jejím zaměřením je vyšší obtížnost úrovní se zajímavým příběhem a vlastní pixel art grafikou. Tato práce bere inspiraci z mé oblíbené hry Super Meat Boy.

Summary

This thesis is about the complete creation of a 2D platforming game created in the Unity game engine. Its focus is on higher difficulty levels with an interesting story and custom pixel art graphics. This work takes inspiration from my favourite game Super Meat Boy.

Čestné prohlášení

Prohlašuji, že jsem předkládanou maturitní práci vypracoval sám a uvedl jsem veškerou použitou literaturu a bibliografické citace.

V Liberci dne

Jan Seidel

Obsah

[Úvod 1](#_Toc155823591)

[1 Použité technologie 2](#_Toc155823592)

[1.1 Vývojové prostředí a programovací jazyk 2](#_Toc155823593)

[1.1.1 Visual Studio 2](#_Toc155823594)

[1.1.2 C# 2](#_Toc155823595)

[1.2 Miro 2](#_Toc155823596)

[1.3 GitHub 2](#_Toc155823597)

[1.4 Unity Engine 3](#_Toc155823598)

[1.5 Aseprite 3](#_Toc155823599)

[2 Unity Engine 4](#_Toc155823600)

[2.1 Prostředí 4](#_Toc155823601)

[2.2 Pojmy 5](#_Toc155823602)

[2.2.1 Game Object 5](#_Toc155823603)

[2.2.2 Asset 5](#_Toc155823604)

[2.2.3 Komponenta 5](#_Toc155823605)

[2.2.4 Transform 5](#_Toc155823606)

[2.2.5 Collider 6](#_Toc155823607)

[2.2.6 Sprite Renderer 6](#_Toc155823608)

[2.2.7 Rigidbody 2D 6](#_Toc155823609)

[2.2.8 Prefab 7](#_Toc155823610)

[2.2.9 Skriptovatelný objekt 7](#_Toc155823611)

[2.2.10 Trigger 7](#_Toc155823612)

[3 Aseprite 8](#_Toc155823613)

[3.1 Prostředí 8](#_Toc155823614)

[3.2 Funkce 8](#_Toc155823615)

[3.2.1 Zobrazení snímků a vrstev 8](#_Toc155823616)

[3.2.2 Onion Skin 9](#_Toc155823617)

[3.2.3 Tilemapy 9](#_Toc155823618)

[3.2.4 Exportování 10](#_Toc155823619)

[4 Vývoj 11](#_Toc155823620)

[4.1 Design 11](#_Toc155823621)

[4.1.1 Vymyšlení příběhu 11](#_Toc155823622)

[4.1.2 Návrh úrovní 11](#_Toc155823623)

[4.1.3 Tvorba grafiky 11](#_Toc155823624)

[4.1.4 Animace 11](#_Toc155823625)

[4.2 Skripty 12](#_Toc155823626)

[4.2.1 Herní mechaniky 12](#_Toc155823627)

[4.3 Sestavení a testování 12](#_Toc155823628)

[Závěr 13](#_Toc155823629)

[Seznam zkratek a odborných výrazů 14](#_Toc155823630)

[Seznam obrázků 15](#_Toc155823631)

[Použité zdroje 16](#_Toc155823632)

[A. Seznam přiložených souborů I](#_Toc155823633)

Úvod

Videohry hraji už od velmi brzkého věku s mým otcem. Nejčastěji to byli různé střílečky z první osoby, jako například Call of Duty nebo Payday 2, dále to byli příběhové hry jako série Uncharted, až po skákačky, jako je Super Meat Boy, a právě díky této hře jsem mi podnítila myšlenka k vytvoření vlastní hry. Další z důvodů je, že už dlouho mě nezaujala žádná nová skákačka.

Většinu svých programovacích zkušeností jsem nasbíral na střední škole. Na základní škole jsme v hodinách informačních technologií zkoušeli pouze úplné základy HTML. Na střední škole jsme se učili od vytváření webových stránek a jejich stylování, základy algoritmizace v Pythonu, desktopové a mobilní aplikace v C#, až po umělé inteligence. Vždy mě, ale nejvíce bavilo hrát hry, a proto během výuky programování ve čtvrtém ročníku během bloku výuky v herním enginu Unity jsem byl velice rád za tuto příležitost a motivaci si naprogramovat vlastní hru.

Nastavte **název** dokumentu a **autora** v nabídce Soubor/Informace.

Pro vkládání zdrojů použijte Reference/Spravovat prameny.

# Použité technologie

## Vývojové prostředí a programovací jazyk

### Visual Studio

Integrované vývojové prostředí (IDE) Visual Studio je kreativní spouštěcí panel, který můžete použít k úpravám, ladění a sestavování kódu a k publikování aplikace. Nad rámec standardního editoru a ladicího programu, které poskytuje většina integrovaných vývojových prostředí (IDE), obsahuje Visual Studio kompilátory, nástroje pro doplňování kódu, grafické návrháře a mnoho dalších funkcí, které zlepšují proces vývoje softwaru.

### C#

C# je moderní, objektově orientovaný a typově bezpečný programovací jazyk. Jazyk C# umožňuje vývojářům vytvářet mnoho typů zabezpečených a robustních aplikací, které běží v .NET. Jazyk C# má své kořeny v rodině jazyků C a programátorům v jazyce C, C++, Javě a JavaScriptu je hned povědomé. C# je objektově orientovaný programovací jazyk orientovaný na komponenty. Jazyk C# poskytuje jazykové konstrukce pro přímou podporu těchto konceptů, díky kterým je jazyk C# přirozeným jazykem pro vytváření a používání softwarových komponent. Od svého původu přidává jazyk C# funkce pro podporu nových úloh a nově vznikajících postupů návrhu softwaru. V jádru je jazyk C# objektově orientovaný.

## Miro

Miro je cloudová aplikace pro týmy nebo jednotlivce, kteří chtějí mít na dálku „velkou tabuli společných nápadů“, jako je běžné ve firmách. Uživatelům umožňuje využití velké plochy pro vkládání obrázku, rozdělování do sekcí pro lepší organizaci, tvořit digitální poznámkové bloky, kreslení a další. Miro je efektivnější způsob organizace práce. Aplikace je dostupná na webu a desktopu.

## GitHub

GitHub je webová služba podporující vývoj softwaru za pomoci verzovacího nástroje Git. GitHub nabízí bezplatný Webhosting pro open source projekty. GitHub je velmi oblíbený mezi vývojáři po celém světě a je považován za jednu z největších služeb pro správu verzí kódu a spolupráci na projektech.

## Unity Engine

Unity Engine je herní engine. Herní engine je vývojové prostředí určené pro tvorbu videoher pro různé platformy, například počítače, herní konzole, mobilní zařízení a další. Unity Engine onsahuje řadu funkcí, které zrychlují a zjednodušují vývoj nových videoher. Poskytují vykreslování předmětů nebo herní fyziku. Dále často obsahují knihovny s připravenými assety, jako jsou různé předměty, modely postav nebo prostředí. To zajišťuje vývojářům příjemnější tvorbu herních světů a prototypů nápadů.

Dalšími herními enginy mimo Unity Engine jsou například Unreal Engine nebo Godot, ale některá herní studia používají vlastní herní engine, jako například Activision Blizzard u Call of Duty. Unity Engine jsem si vybral, protože má dobře vytvořenou dokumentaci a mám zkušenosti z jednoho menšího projektu.

## Aseprite

Aseprite je program pro tvoření pixel-artové grafiky a animací. Je to placený program, který stojí 19.99$. V žebříčkách nejlepších programů pro tvorbu pixel-artové grafiky se umisťuje mezi nejlepší. Sprity jsou tvořeny vrstvy a framy, jako oddělené koncepty. Podporuje velké množství barevných profilů a různých barevných režimů, které jsou RGBA, Indexované (barevné palety pro až 256 barev) a Grayscale. Jednou z výhod tohoto programu je funkce onion skinning, to znamená, že lze zapnout režim ve které lze pozorovat předchozí snímek a díky tomu je snažší tvoření plynulých animací. Animace nebo sprity lze buď ukládat jako .aseprite soubory, ve kterých je uchována časová osa animací, a nebo jako .png soubory.

# Unity Engine

## Obsah obrázku snímek obrazovky, Multimediální software, software, Grafický software Popis byl vytvořen automatickyProstředí

Obrázek Prostředí Unity

Na obrázku výše lze vidět prostředí Unity. Skládá se z podoken, které lze přizpůsobit k vlastním potřebám. Na pravé straně se nachází záložka Inspector, které ukazuje podrobnosti zvolených herních objektů. Například právě teď zobrazuje prvky herního objektu Main Camera: Transform, který určuje polohu, natočení a velikost, Camera, která dělá záznam a zobrazuje herní svět hráčům a Audio Listener, který v podstatě funguje jako mikrofon, protože zaznamenává zvuky vydávané herními prostředky a přehrává je do skutečných reproduktorů. V horní části uprostřed jsou zobrazeny záložky Scene a Game. Tyto záložky přepínají zobrazování mezi hráčským a editorským pohledem. V dolní části uprostřed jsou žáložky Project a Console. Záložka Console se používá pro testování možných funkcí, výpisy chyb a výpisy varováni. V záložce Project jsou zobrazeny všechny používané soubory a celá jejich struktura. Na levé straně obrazovky se nachází záložka Hierarchy, ve které se zobrazují všechny herní objekty nacházející se ve stávající scéně.

## Pojmy

### Game Object

Herní objekt je základním prvkem Unity Enginu reprezentující vše, co se nachází v herním světě. Například postavy, speciální efekty, zvuky, lokace nebo nepřátelé. Samostatný herní objekt, ale nemá žádné vlastnosti, ani funkce. K tomu existují komponenty. Každý herní objekt musí mít komponentu transform.

### Asset

Asset je reprezentací, jakéhokoliv předmětu použitého ve hře nebo projektu. Asset může pocházet ze souboru, který nebyl vytvořen pomocí Unity, jako například 3D model, audio nahrávka, obrázek nebo další typy souborů. Assety vytvořené pomocí Unity mohou být kupříkladu Animator Controller, který mění spuštěné animace podle aktuální situace nebo Render Texture, který může být nedílným nástrojem při ladění efektů, které renderují textury.

### Komponenta

Komponenty dávají funkce herním objektům. Většina komponentů mívá své vlastní atributy určují funkčnost jednotlivých komponent. Jedna z nejdůležitějších komponent v Unity je script. Script dokáže definovat vlastní komponenty, ale i zprostředkovávat a řídit cizí komponenty.

### Transform

Obsah obrázku text, snímek obrazovky, Multimediální software, software

Popis byl vytvořen automatickyNení jediný herní objekt, který by neměl komponentu Transform. Je to komponenta manipulující a ukládající pozici, velikost a natočení herního objektu. Všechny komponenty Transform mohou mít vlastního rodiče, díky kterému mohou aplikovat pozici, velikost a natočení hierarchicky. To je hierarchie, kterou je možné vydět v záložce Hierarchy. Nadále taky podporuje enumerátory a v důsledku jejich použití lze jednoduše procházet jeho potomky.

Obrázek Ukázka Transform komponenty

### Collider

Collider je komponenta sloužící k proměnění ve své podstatě objektu na překážku. Do této překážky lze narážet a pokud je potřeba s touto překážkou hýbat je nutné přidat další komponentu s názvem Rigidbody, díky důsledkem této komponenty je aplikování fyzických vlastností a ovlivnění gravitační silou. Různé skripty na objektu mohou využívat metody „OnCollisionEnter()“, jenž se vyvolá v případě doteku dalšího collideru nebo rigidbody.

### Sprite Renderer

Následující komponenta sprite render se používá k vykreslování objektu na obrazovku. Komponenta může být deaktivována pro zneviditelnění objektu. Jelikož existuje více typů Renderer, ale pro mě jeden z nejdůležitějších je Sprite Renderer. Ten se hlavně používá pro vykreslování spritů a k ovládání jejich vizuálního zobrazení na obrazovce.

### Rigidbody 2D

Obsah obrázku text, snímek obrazovky, software, Multimediální software

Popis byl vytvořen automatickyJedna z nejdůležitějších komponent v Unity je Rigidbody 2D. Rigidbody 2D se používat ke spuštění reagování herních objektů na základě fyziky. Jedním z typů reakce je ovlivnění gravitaci, následují reakcí může být reakce na další tělesa předáním kinetické energie nebo momentum nebo kupříkladu třecí síla.

Obrázek Ukázka Rigidbody 2D komponenty

### Prefab

Prefab je systém umožňující vytvoření, nastavení a uložení herních objektů se všemi komponenty, hodnotami atributů a potomků herních objektů. Prefab se používá jako plánek pro používání herního objektu na více místech ve stejné scéně nebo ve více scénách. Díky prefabům je to lehčí, než zbytečné kopírování a vkládání identických herních objektů, protože prefab umožňuje mít je všechny synchronizované. To znamená, že když změníme cokoliv v prefabu, tak se změna provede ve všech „kopiích“, ale když provedeme změnu v jedné „kopii“, tak se změna stane pouze v ní, a díky tomu lze upravit malé vlastnosti na jednotlivých „kopiích“. Například velikost stromu se může lišit. Také je možnost vnoření prefabů do dalších prefabů, a tím je možné vytvoření komplexní hierarchie objektů, které jsou jednoduché editovat na různých úrovních.

### Skriptovatelný objekt

Skriptovatelný objekt je datový prostor s možným použitím pro ukládání velkéhé množství dat nezávisle od instancí tříd. Jedním z hlavních důvodů použití je pro snížení používání paměti vašeho projektu, tím že se nepoužívají kopie stejných dat. Je to dobré používat u projektů, které používají Prefaby ukládající neměnné data v přiložených MonoBehaviour skriptech. Při každé instanci tohoto prefabu vznikne identická kopie dat a tomu se dá předejít. Na rozdíl od MonoBehaviour skriptů nelze Skriptovatelný objekt připojit k herním objektům a musí být uložen jako Asset v projektu.

### Trigger

Trigger je velice podobný komponentě Collider, poněvadž to je Collider, který neslouží jako překážka, ale jako oblast, při které se volají metody „OnTriggerEnter()“ a „OnTriggerExit()“ pro spuštění skriptů v například důležitých pasážích hry. Výsledkem může být přepnutí hudby během přecházení z jedné lokace do druhé lokace nebo po poražení těžkého nepřítele a dosažení cíle hry.

# Aseprite

## Obsah obrázku snímek obrazovky, text, Multimediální software, Grafický software Popis byl vytvořen automatickyProstředí

Obrázek Prostředí Aseprite

Na ukázce obrázku výše lze vidět rozhraní této aplikace. Skládá se z několika sekcí, které se dají jednoduše přizpůsobit vlastním potřebám. Na levé straně uprostřed je sekce, která obsahuje barevně palety, a to je výběr mezi základními 65 přednastavenými palety, jako je například A64, ZX Spectrum, NES, Game Boy a další s možností si vytvořením vlastní palety. Dále hned pod touto sekcí se nachází výběr barev pomocí RGBA, HSVA, HSLA nebo Gray jako odstín šedé barvy. Hned pod touto sekcí jsou dvě pole, a to barva, která je v popředí a barva, která je v pozadí. Následně se uprostřed obrázku nachází sekce, ve které je kreslící plocha, která lze upravit podle vlastních potřeb. Potom ve spodní sekci uprostřed se nachází, kde jsou jednotlivé snímky animace ve sloupci a jednotlivé vrstvy v řádcích. Na pravé straně je sekce, kde se vybírají funkce, jako je například výběr části, tužka, vymazání, vybrání barvy z už nakresleného obrázku a další.

## Funkce

### Zobrazení snímků a vrstev

Na výše uvedeném obrázku je ve spodní části sekce zobrazující vrstvy a jednotlivé snímky. Snímky, ale i vrstvy se dají kopírovat, a to se u snímků hodí, aby vznikla plynulejší animace, anebo aby animace byla trochu delší. Jednotlivé vrstvy se dají vypínat. Například při tvoření postav je výhodné si rozdělit postavu na jednotlivé končetiny pro rychlejší tvorbu snímků. Může se stát, že například se v daném pohybu nebude pohybovat levá ruka, a tak je jednodušší překopírovat pouze levou ruku. U jednotlivých snímků lze nastavit jejich délka v milisekundách.

### Onion Skin

Obsah obrázku pixel, snímek obrazovky, 3d modelování

Popis byl vytvořen automatickyOnion skin je funkce, která umožňuje si zobrazovat předcházející snímek pro přesnější kreslení snímku dalšího. Lze si vybrat libovolnou velikost počtu snímků, takže je možné například pozorovat 5 snímku zpět. Tato funkce je jedna z nejlepší na tomto programu a z veliké časti to je můj důvod výběru této aplikace.

Obrázek Ukázka funkce Onion Skin

### Tilemapy

Tilemapy jsou speciální druh vrstev, kde každá buňka plochy na kreslící ploše je referencí na tile v tilesetu. Tile je malý obrázek, který může být použitý na více místech v Tilemap vrstvě. Tileset je kolekce tilů o stejné velikosti. Tilemap vrstva je 2D obrázek, Obsah obrázku vzor, zelené

Popis byl vytvořen automatickykde každý pixel má index pole, který je referencí na určitý tile v tilesetu.

Obrázek Ukázka Tilesetu země v lesní části hry.

### Exportování

Exportování je nedílnou součástí tvoření pixel artu, protože k čemu je vytvořený pixel art, který nemůžeme nikde jinde použít. Animace se dají exportovat jako .gif soubor nebo jako sprite sheet. Lze exportovat, třeba jen určitou část kreslící plocha, anebo jen jednotlivé vrstvy nebo snímky animace. Sprite sheet je typ exportu, kde se jednotlivé snímky animace dají za sebe do dlouhé řady a tím vznikne .png soubor.   
V samotném unity existuje balíček, který umožňuje export rovnou .aseprite souborů a tím urychluje nahrávání animací, protože místo tvoření animace z několika snímku a nastavování časové osy to ten balíček udělá za vás podle nastavení v asepritu.

Obrázek Ukázka Sprite Sheetu animace smrti

# Vývoj

## Design

### Vymyšlení příběhu

Jako milovník žánru fantasy jsem vždy chtěl vytvořit si vlastní příběh, který bude reprezentovat moji představivost. Proto jsem se rozhodl o vytvoření středověkého světa s nadpřirozenými prvky a na konci s manipulací časoprostorem. Cílem příběhu hry je snaha získání vlastní svobody a následný závěr je důsledkem chamtivosti a touhy po něčem větším. Scénář ke hře je přiložen jako soubor. Hlavní postava Sir Aric je zobrazována jako šlechetný, avšak nemrtvý rytíř, který má silný smysl pro věc. Je odhodlaný splnit svůj úkol a je mu jedno co ho to bude stát. Morgana, která vyvolala Sira Arica, je tajuplná a velice rozzuřená čarodějnice, která tráví svůj život v temném lesním doupěti plánováním tím, jak sebrat nesmírně silný krystal od krále Rolanda. Král Roland královský, ale neklidný panovník, je autoritou s historií vyrytou do vrásek jeho znavené tváře.

### Návrh úrovní

### Tvorba grafiky

Jako někdo, kdo v podstatě nikdy netvořil grafiku, jsem se rozhodl, že si to chci vyzkoušet a nasbírat zkušenosti. Proto mě napadlo, že když budu muset vytvořit celý pixel art styl pro vlastní hru, tak nebudu moci použít existující assety a vše si budu muset vytvořit sám. To je ohromná motivace se tomu začít věnovat. Pro pixel art styl jsem se rozhodl, protože pixelová mřížka poskytuje strukturovaný systém, který pro mě jako nováčka v grafickém designu je snadněji pochopitelný. Také dalším důvod výběru pixel art stylu je retro estetický vzhled. Jednoduchost pixel artu může evokovat pocit klasických videoher a poskytnout hře jedinečný a vizuálně působivý vzhled.

### Animace

Když už se vytváří pixel art grafika u hry, tak je samozřejmostí, že bude potřeba také tvořit vlastní animace. Samotné nízké rozlišení pixel artu zjednodušuje proces tvoření animací a usnadňuje ho začátečníkům, jako jsem já, pro které by tradiční animační techniky mohli být příliš obtížné.

## Skripty

### Herní mechaniky

## Sestavení a testování

Závěr

Tak jsem se dostal až na konec.

Seznam zkratek a odborných výrazů

Engine

Je v informatice označení pro ústřední část softwarového produktu, v kontextu videoher je to platforma, která poskytuje nástroje pro tvorbu her

Seznam obrázků

Použité zdroje

1. ***Miro.com.* [Online] [Citace: 15. 11 2023.] https://miro.com.**

**2. Wikipedie, Přispěvatelé. Unity (herní engine). *Wikipedie: Otevřená encyklopedie.* [Online] 12. 11 2023. [Citace: 15. 11 2023.] https://cs.wikipedia.org/w/index.php?title=Unity\_(hern%C3%AD\_engine)&oldid=23372746.**

**3. Igara Studio. Aseprite Docs. *Aseprite.* [Online] 28. 12 2023. [Citace: 28. 12 2023.] https://www.aseprite.org/docs/.**

**4. Unity Technologies. Unity Documentation. *Unity.* [Online] 13. 11 2023. [Citace: 15. 11 2023.] https://docs.unity3d.com/Manual/index.html.**

**5. GitHub, Inc. Github Docs. *Github.* [Online] 28. 12 2023. [Citace: 28. 12 2023.] https://docs.github.com/en.**

1. Seznam přiložených souborů

Na přiloženém datovém nosiči se nacházejí následující soubory a složky:

* **MP2010-Novák-Jan-L4-Tepelné\_čerpadlo.docx** – editovatelná verze dokumentace maturitní práce
* **MP2010-Novák-Jan-L4-Tepelné\_čerpadlo.pdf** – tisknutelná verze dokumentace maturitní práce
* **Výkresy** – kompletní výkresová dokumentace
* **Aplikace** – zdrojové kódy